UART唤醒并且频率补偿

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

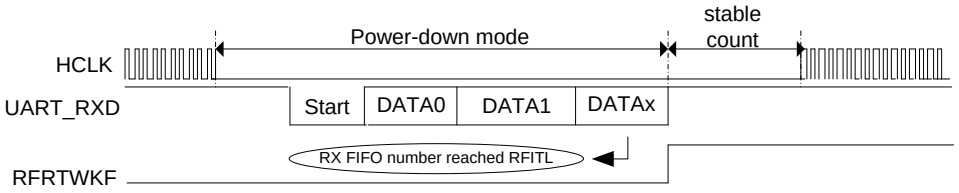
代码简述	本范例代码使用UART把M480从低功耗模式唤醒并进行频率补偿
BSP 版本	M480 Series BSP CMSIS V3.04.000
开发平台	NuMaker-PFM-M487 Ver 3.0

1 功能介绍

1.1 简介

此范例程序中，M480会进入power down模式，当所收到UART的数据达到门坎值，即会唤醒M480。由于power down模式下 UART 时钟源需要设为 LXT，因此使用UART鲍率补偿以达到更准确的频率。使能缓存器WKRFRTEN(UART_WKCTL[2])可开启UART唤醒功能。

进入power down模式后，当RX FIFO里接收到的数据达到了门坎值设定 RFITL (UART_FIFO[7:4])，会触发接收数据标志 RFRTWKF(UART_WKSTS[2])。下图是UART接收数据达到门坎值唤醒示意图：



1.2 原理

使用者可以在BRCOMP(UART_BRCOMP[8:0])设定需要补偿的位，并且在BRCOMPDEC(UART_BRCOMP[31])设定正补偿或是负补偿，使所设定的位数据时间增加或减少1个UART的频率。以下为缓存器说明：

Bits	Description	
[31]	**BRCOMPDEC**	**Baud Rate Compensation Decrease** 0 = Positive (increase one module clock) compensation for each compensated bit. 1 = Negative (decrease one module clock) compensation for each compensated bit.
[30:9]	**Reserved**	Reserved.
[8:0]	**BRCOMP**	**Baud Rate Compensation Patten** These 9-bits are used to define the relative bit is compensated or not. BRCOMP[7:0] is used to define the compensation of UART_DAT[7:0] and BRCOM[8] is used to define the parity bit.

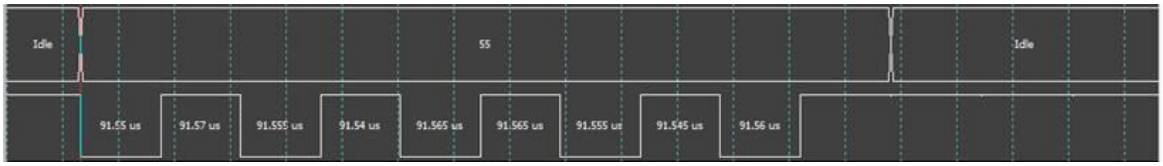
范例如下：

UART peripheral clock = 32.768K (LXT)

UART Baud rate = 9600, 1bit = 104.167us

原波形会产生3个UART频率的时间来表示1个位的UART数据 ((1 / 32.768K) * 3 = 91.55us)，

波形如下：



设定补偿缓存器如下：

BRCOMP[8:0] = 0x0010100101

BRCOMPDEC[31] = 0

因此在bit0, bit2, bit5, bit7会补偿1个UART频率的时间 = ((1 / 32.768K) * (3+1) = 122.07us)

波形如下：

开启补偿功能可以控制UART鲍率的误差在± 0.5个UART频率的时间以内。

2 代码介绍

首先设定UART1的传输速率为9600，并且对bit0, bit2, bit5, bit7做频率补偿。接着开启UART唤醒功能，再来设定UART接收数据的门阀值。其程序代码如下：

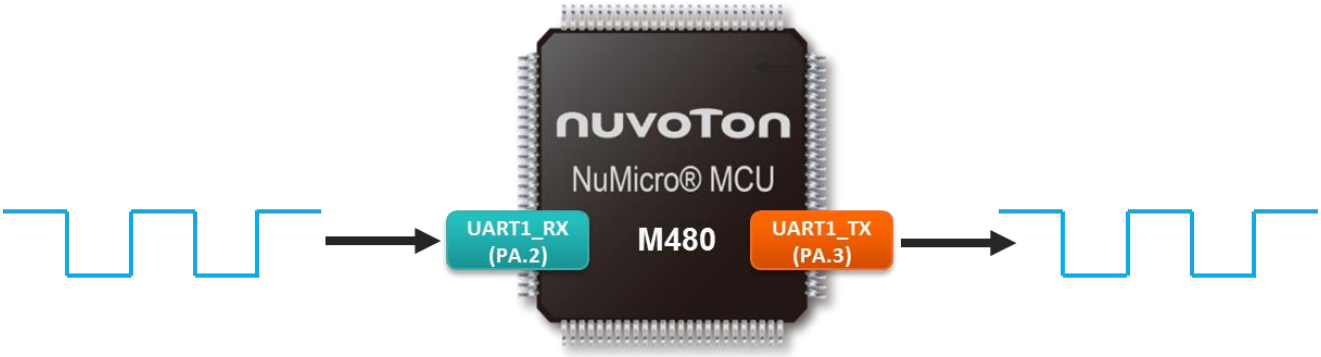
```
void UART_FunctionTest()
{
    SYS_UnlockReg();

    /* Configure UART1 and set UART1 baud rate */
    UART_Open(UART1, 9600);

    /* Set Compensation in bit0, bit2, bit5, bit7 */
    UART1->BRCOMP = 0xA5;

    g_bWait = 0;

    UART1->WKCTL = UART_WKCTL_WKRFRTEN_Msk;

    /* Enable UART RDA/THRE/Time-out interrupt */
    NVIC_EnableIRQ(UART1_IRQn);
    UART_EnableInt(UART1, (UART_INTEN_WKIEN_Msk));

    CLK->PWRCTL |= CLK_PWRCTL_PDWKIEN_Msk;
    NVIC_EnableIRQ(PWRWU_IRQn);

    /* clear status */
    UART1->WKSTS = UART1->WKSTS;

    /* Wait debug message finish */
    while((UART0->FIFOSTS & UART_FIFOSTS_TXEMPTYF_Msk) == 0);

    UART1->FIFO &=~ UART_FIFO_RFITL_Msk;
    UART1->FIFO |= UART_FIFO_RFITL_4BYTES;

    /* Enter to Power-down mode */
    CLK_PowerDown();

    while(!g_bWait);
    CLK_SysTickLongDelay(1000000);
```

```
    /* Disable UART RDA/THRE/Time-out interrupt */
    UART_DisableInt(UART1, UART_INTEN_WKIEN_Msk);
    g_bWait = TRUE;

    printf("\n\n%x\n",UART_READ(UART1));
    printf("\n\n%x\n",UART_READ(UART1));
    printf("\n\n%x\n",UART_READ(UART1));
    printf("\n\n%x\n",UART_READ(UART1));

    printf("\n\nUART Wakeup Compensation Sample Demo End.\n");
}
```

3 软件与硬件环境

- 软件环境
  - BSP 版本
    - M480 Series BSP CMSIS V3.04.000
  - IDE 版本
    - Keil uVersion 5.26
- 硬件环境
  - 电路组件
    - NuMaker-PFM-M487 or other M480 Development Board
  - 示意图

从 UART1_RX(PA.2)输入讯号，当数据达到门坎值时会唤醒 M480。



4 目录信息

- EC_M480_UART_Wakeup_Compensation_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 UART_Wakeup_Compensation.uvproj
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
Oct. 1, 2019	1.00	1. 初始发布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.